SIMple ID: QR Codes for Authentication using Basic Mobile Phones in Developing Countries

Chris Hicks[1], Vasilios Mavroudis[1], and Jon Crowcroft[1][2]

[1] The Alan Turing Institute
{c.hicks, vmavroudis, jcrowcroft}@turing.ac.uk
[2] The University of Cambridge
jon.crowcroft@cl.cam.ac.uk

Abstract. Modern foundational electronic IDentity (eID) systems commonly rely on biometric authentication so as to reduce both their deployment costs and the need for cryptographically capable end-user devices (e.g., smartcards, smartphones). However, this exposes the users to significant security and privacy risks. We introduce SIMple ID which uses existing infrastructure, Subscriber Identity Module (SIM) cards and basic feature phones, to realise modern authentication protocols without the use of biometrics. Towards this goal, we extend the international standard for displaying images stored in SIM cards and show how this can be used to generate QR codes on even basic no-frills devices. Then, we introduce a suite of lightweight eID authentication protocols designed for on-SIM execution. Finally, we discuss SIMple ID's security, benchmark its performance and explain how it can enhance the security and privacy offered by widespread foundational eID platforms such as India's Aadhaar.

Keywords: Authentication, Identity Management, Trusted platforms

1 Introduction

More than 60 less-developed countries have launched national foundational identity programs in the last 15 years [42]. Unlike functional identity, which claims specific attributes about people such as voting entitlement or drivers' licensing, foundational identity is principally concerned with asserting the uniqueness of each person [22]. In practice, often because there is no reliable civil registry to bootstrap, the uniqueness of each resident is determined using biometric deduplication during enrollment. India's Aadhaar platform for example, which has generated more than 1.3 billion unique foundational identities [11], requests samples of all ten fingerprints, both irises and a portrait photograph during enrollment. Aadhaar and the Modular Open Source Identity Platform (MOSIP) (i.e., "Aadhaar in a box" [64]) are already being trialed and adopted in six different countries [12]. They have successfully proven the foundational identity model for development and are set to impact the lives of many millions more in the near future.

Aadhaar, MOSIP and other foundational electronic IDentification (eID) platforms must provide digital authentication mechanisms that extend access to government-subsidised goods and services, as well as opportunities to build credit and reputation, to some of the poorest in society [43]. The design of authentication mechanisms is hence constrained by less-developed infrastructure, including domestic power and mobile network coverage, and limited access to expensive technologies such as smartphones. Whilst smart cards have been widely adopted for authentication in developed countries [69] the same is not true globally. Developing countries are more likely to find the additional capital, skilled labour and infrastructure required by smart cards prohibitive [5, 77]. Aadhaar does not issue a smart card and, of the approximately 72 billion transactions processed to date, over 76% were based on biometric authentication. Most of the remaining transactions were authenticated by submitting sensitive personal information, such as a name and address (i.e., "demographic authentication"). In total, less than 4% of Aadhaar transactions were authenticated using One-Time-Passwords (OTPs) sent to resident's mobile phones using Short Message Service (SMS) [14].

Contemporary foundational eID systems are not making good use of basic mobile phones, even though such devices are more common than smartphones in many less-developed countries [18]. This is mostly due to the limited capabilities of basic phones (e.g., lack of secure enclaves [10]). SIMple ID is a mobile identity solution that overcomes these limitations. Based on the internationally standardised mechanism for displaying images stored on a Subscriber Identity Module (SIM) and the cryptographic capabilities of SIM cards, SIMple ID uses QR codes to transform low-cost mobile handsets into secure authentication credentials.

Contributions

1. We propose a practical extension to the international mobile communication standards that provides a secure authentication mechanism designed for the unique sociotechnical landscape of less-developed countries.
2. We evaluate our full, open-source implementation of SIMple ID[1] which includes a Java Card applet and a KaiOS patch implementing the SIM and mobile handset parts of our protocol, respectively.
3. Beyond authentication, SIMple ID provides a robust general mechanism for establishing a QR code channel between a SIM and an in-person verifier; enabling other applications such as digital payments.

2 Preliminaries

Here we provide the prerequisites for a complete understanding of SIMple ID.

2.1 Mobile phones in developing countries

SIMple ID is motivated to use basic phones for authentication because they are still widely used in a number of countries. On a global scale, basic phones

[1] `https://github.com/alan-turing-institute/simple-id`

accounted for 32% of total mobile connections in 2020 [29]. Narrowing down to the Sub-Saharan Africa region that percentage rises to 52%. In 2020, 46% of its 1.1 billion inhabitants [17] were mobile network subscribers meaning over 250 million connected using a basic device. Moving from Africa to India, whilst mobile broadband covers 99% of the country and 77% of mobile subscribers now have access to a 4G-capable handset; 49% of adults still had not adopted a smartphone in 2020 [29].

2.2 UICC and (U)SIM cards

Beginning with 2G Global System for Mobiles (GSM) networks, Subscriber Identity Module (SIM) cards; which are based on smart cards, provided a portable mechanism for subscriber identification and authentication [47]. Originally denoting a unified hardware and software package, starting with 3G standards the two parts were separated and renamed. SIM hardware is now termed a universal integrated circuit card (UICC) [30] and is defined by standards including ISO 7816 [53] and ETSI TS 102 221 [33]. UICCs are low-cost, and low-performance, but generally provide a certified degree of tamper-resistance [48, 74]. UICCs have their own Central Processing Unit (CPU), Read-Only Memory (ROM), Electrically Erasable Programmable ROM (EEPROM) and often include a dedicated cryptographic co-processor. All UICCs support the cryptographic algorithms needed for mobile network authentication including secure random number generation, cryptographic hashing and symmetric encryption. Many UICCs also support public-key algorithms such as RSA, DSA, ECDH and ECDSA [49, 78].

A UICC always run at least one Network Access Application (NAA) such as the SIM [39] and Universal SIM (USIM) applications [37] for connecting to GSM and 3G+ networks, respectively. Modern UICCs are based on the Java Card Platform (JCP) which provides a Java Card Runtime Environment (JCRE), an Application Programming Interface (API) and a Java Card Virtual Machine (JCVM) [56]. The JCRE is the operating system of a Java Card and manages the shared facilities including the communication protocols, channels, interrupts, access conditions, applications and files. All Java Card UICCs also follow the GlobalPlatform standards [2] which define an API for supporting multiple applications and managing their life-cycles independently [30]. In addition, Java Card UICCs support Over-The-Air (OTA) remote management of applications and files by the network operator [35].

UICC file system UICCs have an elementary file system based on a hierarchical file structure that is used to manage the life cycle of all applications and data on the Java Card [33]. As shown in Fig. 1, the root of the file structure; termed the Master File (MF), is home to a number of subdirectories called Dedicated Files (DFs), application-specific subdirectories known as Application Dedicated Files (ADFs) and leaf nodes termed Elementary Files (EFs) which contain only data. The UICC API supports creating, deleting, (de)activating, reading, updating and resizing files on the UICC programmatically at run-time using an application with appropriate privileges.

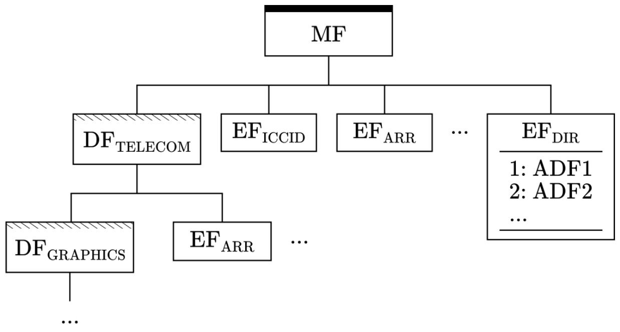

Fig. 1: The hierarchical UICC file and application structure.

Card Application Toolkit The Card Application Toolkit (CAT) generically defines the interface between the UICC and the Mobile Equipment (ME) host (e.g., mobile phone) [34]. The CAT is commonly referred to as the SIM Toolkit (STK) [32] which more specifically designates the CAT subset supported by the GSM SIM application [39]. Similarly, the USIM Application Toolkit (USAT) [38] refers to the CAT subset supported by the 3G+ USIM application [37]. In addition to providing the UICC-ME interface which is needed for basic GSM and 3G+ network access, the CAT also provides a mechanism for the ME user to interact with the UICC through a basic menu system.

A key feature of the CAT is that it allows the UICC to proactively initiate actions taken by the ME. Since the standard smart card API is based on the model of an active host controlling a subordinate smart card [53], this is achieved through regular polling of the UICC by the ME. Proactive UICC commands for user interaction include setting up a menu, displaying text, playing tones, retrieving user input and launching the web browser [34]. Proactive UICC commands are commonly used for providing subscriber services including value-added operator content [75], mobile money [73] and digital signatures [62].

2.3 QR codes

QR codes are an internationally standardised [46, 52] form of barcode that, although originally intended to support traceability in automotive supply chains [1], have been widely adopted by mobile applications including digital identification [9, 71] and payments [79]. QR codes are configured using 40 different version numbers and 4 different levels of error correcting code (ECC). Each version number prescribes a specific size, form and data capacity; meanwhile, the level of ECC determines the tolerance a code has to obscuration and physical damage. Each QR code version has a specific number of modules, ranging from 21×21 for version 1 to 177×177 for version 40, which also determine the minimum number of visual elements needed to draw the barcode. In other words a version 1 QR code requires at least 21×21 pixels on a screen, or dots on a page, for its construction. The ISO standard [52] further specifies an additional 4 modules should be left blank around the QR code on all sides to ensure readability.

2.4 Cryptographic primitives and notation

SIMple ID uses standardised cryptographic primitives from open standards. The use of Java Card UICCs considerably limits the scope of primitives to traditional and well-analysed algorithms [78]. The following high-level presentation aims foremost to document our notation for the familiar reader and secondly, to provide an intuition and references for the unacquainted.

Encryption schemes, used to ensure the secrecy of information, are a triple $(\textsc{Gen}, \textsc{Enc}, \textsc{Dec})$ of probabilistic polynomial time (PPT) algorithms for key generation, encryption and decryption, respectively. Where η is a security parameter, $\textsc{Gen}$ outputs a pair of bit strings (e, d) such that $\forall (e, d) \leftarrow \textsc{Gen}(1^\eta)$, and for every message $m \in \{0,1\}^\star$, $Pr[\textsc{Dec}_d(\textsc{Enc}_e(m)) = m] = 1$. In private key schemes $d = e$ whereas for public key schemes $d \neq e$, and e is termed the public key which we denote P_e. The private key d is denoted k_d. Informally an encryption scheme provides (semantic) security when, in the absence of k_d, encrypted messages tell an adversary nothing about the original message [45].

Digital signatures provide assurances about the authenticity of data and comprise a triple of PPT algorithms $(\textsc{Gen}, \textsc{Sign}, \textsc{Ver})$ for key generation, signing and verification, respectively. $\textsc{Gen}$ outputs a pair of bit strings (k_s, P_v) such that $\forall (k_s, P_v) \leftarrow \textsc{Gen}(1^\eta)$, and for every message $m \in \{0,1\}^\star$, $Pr[\textsc{Ver}_{P_v}(\textsc{Sign}_{k_s}(m)) = 1] = 1$. Informally, secure digital signature schemes demand that no adversary can forge even a single valid signature on any arbitrary message [45].

One-time passwords are a popular mechanism for authentication often encountered as a second-factor when using online password-based systems. One of the most widely used algorithms is the HMAC-Based OTP (HOTP) first described in 2005 complete with an analysis of its security [28]. In the rest of this work we use $\mathrm{HOTP}_{k_{\mathrm{OTP}}}(c)$ to denote the algorithm described in the standard as parameterised by private key k_{OTP} and counter c which are shared and synchronised between the client and server.

3 The foundational eID model

Here we introduce the authentication and adversary models of foundational eID. Aadhaar, and by extension MOSIP [67], is used as the exact basis because it is more widely described in the literature than other foundational eID systems [8, 20, 71, 72]. First we provide a brief explanation of the enrolment process before outlining the authentication and threat models that guide our development.

Regardless of the specific implementation, foundational eID is based on a Centralised IDentity Repository (CIDR) which is populated by a continuous enrollment process. Residents with identities in the CIDR are provided a Unique IDentity (UID) number that, used alongside one or more authentication factors, is used to prove identity. Enrollment is designed to ensure a one-to-one correspondence between enrolled residents and unique digital identities. Biometric enrollment, which is required to provide strong one-person-one-identity guarantees, necessitates a centralised architecture that can pairwise compare each new enrollment with the identities already in the system. In addition to collecting biometric data, including fingerprint and iris scans; enrollment also captures personal demographic information such as name, address and date of birth [6, 16].

3.1 Authentication

Authentication in the foundational paradigm requires a UID, or rarely a Virtual ID (VID), along with one or more authentication factors. The process usually takes place in-person, between a resident and a requesting entity, and a fingerprint scan is used for verification [14]. The CIDR is queried in every authentication as the scanned fingerprint biometric must be evaluated against the fingerprints associated with the submitted UID during enrollment. If the fingerprint matches, or no match is found, the CIDR returns a digitally signed "yes" or "no" response to the requesting entity, respectively. The fingerprint can be substituted with, or complemented by, the iris biometric, demographic or mobile OTP factors depending on the residents assets and the specific assurances required.

The full Aadhaar authentication ecosystem involves service agencies (SA) and user agents (UA) that form a distributed network of secure channels from the CIDR to requesting entities. In our general model of foundational authentication shown in Fig. 2, a resident R authenticates to the requesting entity RE by submitting their UID and one or more authentication factors. The RE, using a certified software and hardware stack [4], composes a digitally signed and encrypted Personal Identity Data (PID) block from the factors. Next, the *RE* sends the UID, and the encrypted PID, to the CIDR using the SA-UA network. The CIDR validates and decrypts the PID and then, using the UID to identify the correct record, checks whether the authentication factors in the PID are a match with the stored values. The CIDR digitally signs the "yes" or "no" response and returns it to the RE over the SA-UA network. Finally, the RE's authentication software validates the response and indicates the result.

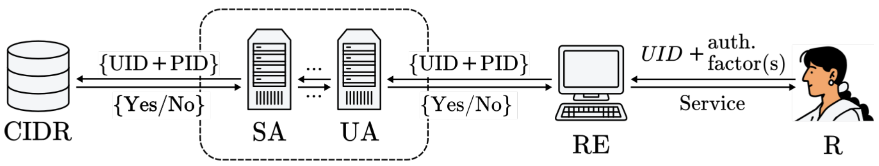

Fig. 2: The foundational eID authentication model.

3.2 Threat model

Making no assumptions about residents access to documentation or technology means it is not practically feasible [40] to ensure uniqueness, or allow for digital authentication, without placing a lot of functionality and trust in a centralised architecture. The CIDR collects the personal and biometric information collected during enrolment and also the authentication data for every transaction. In addition, Aadhaar's SAs and UAs maintain a time-limited log of each authentication including the UID, the encrypted PID block, the CIDR's "Yes/"No"response and any other information (e.g., e-KYC or age-verification) returned upon successful authentication [71]. REs learn all of the authentication data during every transaction except for the authentication factors encrypted in the PID. Aadhaar specifies data retention policies, access controls, biometric hardware standards and auditing practices that aim to avoid or detect vulnerabilities and inappropriate data collection. Unsurprisingly, a recent analysis identified several high-impact security and privacy breaches including subsets of the CIDR being made public and insiders making unauthorised changes to the CIDR [71]. To summarise, the foundational eID adversary model assumes the CIDR, the SA-UA network operators and the REs operate in the semi-honest model [45] by correctly executing the protocols but are able to keep a record of the intermediate computations.

4 CAT QR codes

This section describes our technique allowing basic mobile phones to display full-screen QR codes filled with arbitrary UICC data. As described in Sect. 4.1 several proactive UICC commands support the inclusion of a graphical icon that, at the programmers discretion, can replace an otherwise text-based notification to the user. Our mechanism builds upon this standardised functionality [34], which remarkably already allows QR codes stored on the UICC as native CAT icons to be displayed using some unmodified phones, with a new image coding scheme that supports rendering image file data as a QR code.

4.1 Native icon protocol

A little-known feature of the CAT allows a subset of the proactive UICC commands to include an icon, such as those shown in Fig. 3, intended to enhance the user experience by providing graphical information. The UICC can also request that the icon should entirely replace the text that would otherwise be shown [34]. Icons have been supported by UICCs since GSM STK standards, albeit optionally for ME, but to the best of the authors' knowledge have received little academic or real-world attention. UICC icons support three proprietary coding schemes: black-and-white, 8 bit colour and colour-with-transparency, and are not limited in the standards to any maximum dimensions or file size [37, 39].

The standardised CAT protocol for displaying a native icon, shown in Fig. 4, involves using the `DISPLAY TEXT` proactive UICC command [34] that supports displaying text or an icon to the screen. To show an icon the command must include an icon qualifier bit, indicating whether the icon is “self explanatory”, and an icon identifier specifying which icon to display. Self explanatory icons replace the text usually displayed by the command, potentially providing more

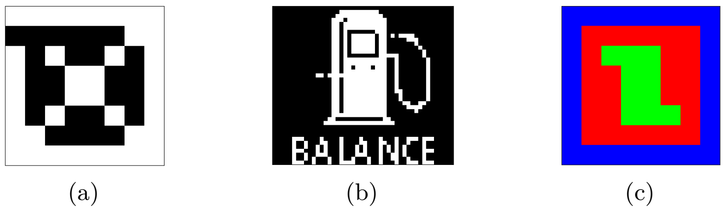

Fig. 3: Three exemplar UICC icons from the ETSI CAT conformance specifications [36]. Icons (a) and (b) use the basic, black and white coding scheme whilst icon (c) uses the colour coding scheme and a three colour palate.

screen space for the icon. It must be noted that support for displaying icons is optional and the phone can choose to ignore the icon identifier. The terminal response may indicate to the UICC when this occurs.

Icons are selected by double reference. First, the icon identifier specifies a record number in a special lookup table called $\mathrm{EF}_{\mathrm{IMG}}$; where the width, height, coding scheme and FID of the icon is stored. Only after $\mathrm{EF}_{\mathrm{IMG}}$ has been read can the icon be retrieved by the phone using the FID. Reading files from the UICC, including $\mathrm{EF}_{\mathrm{IMG}}$, is handled by the phone using ISO 7816-4 commands [53]. If the icon file is bigger than 256 bytes, the standardised maximum response payload, then it is read sequentially in chunks of 256 bytes or less. Icon files must be stored in the $\mathrm{DF}_{\mathrm{GRAPHICS}}$ subdirectory (see Fig. 1) with an FID `4FYY` where `YY` is between 0 and `FF`.

Once the icon file has been read by the phone, the coding scheme byte specified by the corresponding $\mathrm{EF}_{\mathrm{IMG}}$ record is used to render the icon to the screen. The coding scheme byte is either $0x11$, $0x21$ or $0x22$ corresponding to black and white, colour and colour with transparency respectively. The coding schemes are all proprietary formats that begin with the width and height, optionally include colour metadata and then contain the image data using one (or several) bits per image raster point [37].

4.2 QR code rendering

Extending the native icon protocol to efficiently support QR codes is a relatively straightforward task that only requires defining a new image coding scheme. We do just this with a new coding scheme we term 'Render as a QR code' and assign the byte value `0x31`. The protocol for rendering a QR code is unchanged from the native one described in the previous section except for the following:

1. $\mathrm{EF}_{\mathrm{IMG}}$ records corresponding to QR code data must use the new coding scheme byte value.
2. Icon files no longer store image metadata and raster points but instead contain the data that will be stored in the QR code.
3. The mobile phone must implement the new coding scheme with a standard QR code rendering functionality.

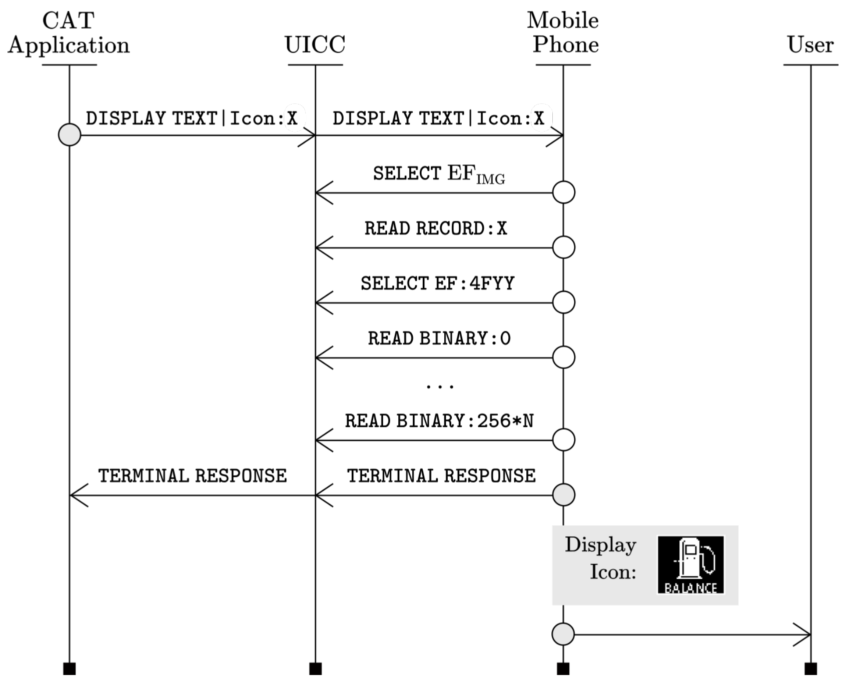

Fig. 4: The standardised CAT protocol for displaying a native icon. The application specifies the icon using the $\mathrm{EF}_{\mathrm{IMG}}$ record index `X` that maps to the icon `YY` with FID `4FYY`. Large icons are split into `N` chunks each 256 bytes or less.

Note that the changes do not deviate significantly from the existing standard. Manufacturers can thus easily implement the new icon protocol without having to allocate significant development resources. This is a significant advantage of the proposed scheme as it makes widespread adoption easier.

5 SIMple ID

Building upon our technique for displaying QR codes on basic phones, we introduce two authentication protocols in the standard foundational eID model. In particular we use the model from Sect. 3.1 but for generality, and ease of presentation, treat the SA-UA network as a secure channel. We evaluate this assumption further in Section Sect. 6.1. Both of our protocols improve upon the security and privacy offered by current foundational eID platforms. Our first protocol lightly builds upon the OTP authentication already used in Aadhaar. We move the OTP generation onto the UICC and display it along with the resident's UID in a low-version QR code. Our second protocol uses public key cryptography to provide enhanced privacy from prying requesting entities.

Firstly, our protocols assume that the CIDR issuer I has securely generated two public key pairs $(P_{I\mathrm{sig}}, k_{I\mathrm{sig}})$ and $(P_{I\mathrm{enc}}, k_{I\mathrm{enc}})$ for digital signatures and encryption, respectively. These keys should be generated according to the standard guidelines [7, 57] noting the constraints of cryptography supported by Java Card UICCs [66, 78]. The parameters HOTP, including the number of OTP digits $d \geq 6$, synchronisation parameter s and throttling parameter t, should also be chosen as indicated in the respective standard [28]. We define I's public keys $P_{I\text{sig}}$, $P_{I\text{enc}}$ and d as public parameters and therefore, to improve readability, do not explicitly specify them as inputs. Lastly, we assume the resident R is already enrolled in the foundational eID platform run by the issuer I and is able to authenticate using one of the usual biometric or demographic mechanisms. Both of our protocols share a common setup and personalisation phase as follows.

SIMple-Setup is run between the UICC U and the issuer I and assumes a secure channel between them. This phase is run only once and can take place either during manufacture or OTA, although the latter requires providing U with I's public keys during manufacture. This is common practice with smart card eID solutions although care must be taken not to repeat past mistakes [70]. At the beginning, I has private keys $k_{I\text{sig}}$, $k_{I\text{enc}}$ and U has unique ICCID *icc-id*.

1. I sends the public keys $(P_{I\text{sig}}, P_{I\text{enc}})$ and d_{OTP} to the UICC U.
2. U generates a uniform random secret key k_{OTP} and initialises a non-secret counter $c := 0$. U also generates two public key pairs, $(P_{U\text{sig}}, k_{U\text{sig}})$ and $(P_{U\text{enc}}, k_{U\text{enc}})$, for digital signatures and encryption, respectively.
3. U sends public keys $(P_{U\text{sig}}, P_{U\text{enc}})$, ICCID *icc-id*, and OTP parameters (k_{OTP}, c) to I. These values are stored as a single record by I for association with the UID of a specific resident in the next phase.

SIMple-Personalise bootstraps the resident's existing foundational authentication credential(s) to securely link their UID with the UICC U. Recall from Sect. 3.1 that the authentication credential is securely signed and encrypted, for the issuer I, using certified hardware and software provided to the requesting entity V. This phase is run between the resident R, U, V and I. Personalisation takes place after R receives the UICC and only needs to succeed once. This can be adjusted in favor of personalisation during OTP-Setup, however SIMple-Personalise allows UICCs to be quickly and widely distributed using e.g., already well-established networks of mobile agents [54]. To begin, R has UID *uid* and U has signing key $k_{U\text{sig}}$, encryption key $k_{U\text{enc}}$ and ICCID *icc-id*. The issuer I has signing key $k_{I\text{sig}}$, encryption key $k_{I\text{enc}}$, the CIDR containing every unique residents' UID *uid* and records of all $(P_{U\text{sig}}, P_{U\text{enc}})$, ICCID *icc-id* and OTP parameters (k_{OTP}, c) submitted in the SIMple-Setup phase.

1. The resident R generates a uniformly random Personal Identification Number (PIN) *pin* and sends the UID *uid* and *pin* to U using their mobile phone.
2. The UICC U generates a uniformly random d-digit session identifier *sid*, signs the personalise message $m_{\text{pers.}} = \textsc{Sign}_{k_{U\text{sig}}}(\text{"I"} | \textit{icc-id} \parallel \textit{uid} \parallel \textit{sid})$ and then encrypts the personalise ciphertext $c_{\text{pers.}} = \textsc{Enc}_{P_{I\text{enc}}}(m_{\text{pers.}})$.

3. R sends the encrypted authentication factor $c_{\text{auth.}}$ (i.e., the regular biometric or demographic PID block) and the personalise ciphertext $c_{\text{pers.}}$ (e.g., shown as a QR code using a basic phone) to the requesting entity V.
4. V sends $c_{\text{auth.}}$ and $c_{\text{pers.}}$ to the issuer I (e.g., using the SA-UA network).
5. I decrypts the personalise ciphertext to recover the personalise message $m_{\text{pers.}} = \textsc{Dec}_{k_{I\text{enc}}}(c_{\text{pers.}})$ containing *icc-id*, *uid* and *sid*. I uses *uid* to find the corresponding CIDR eID record and then verifies the regular PID $c_{\text{auth.}}$ using the standard process.
6. If this first verification succeeds (e.g., R's fingerprint matches the fingerprint(s) in the CIDR record corresponding to the *uid* submitted in Step 1.), then I verifies the UICC signature on $m_{\text{pers.}}$ using the public key ($P_{U\text{sig}}$ linked to *icc-id* in the SIMple-Setup phase. Only if both of these verifications succeed will I tentatively link the UICC's *icc-id*, public keys $(P_{U\text{sig}}, P_{U\text{enc}})$ and OTP parameters (k_{OTP}, c) to the resident R's UID.
7. If the verifications succeeded then I signs the response message $m_{\text{resp}} = \textsc{Sign}_{k_{I\text{sig}}}(``yes'' \parallel sid)$ or else $m_{\text{resp}} = \textsc{Sign}_{k_{I\text{sig}}}(``no'')$. I sends m_{resp} to the requesting entity V.
8. V verifies the signature on m_{resp} and, if successful and m_{resp} contains the session identifier sid', sends sid' to the resident R. If unsuccessful then the protocol terminates and must begin again from Step 1.
9. R sends sid' to the UICC U using their mobile phone. If $sid' = sid$ then the resident's UID *uid* and PIN *pin* are stored by U in non-volatile memory. If unsuccessful after several attempts, the protocol terminates and must begin again from Step 1.

At this stage the UICC has been personalised for a specific resident and our two authentication protocols differ in the final phase as now described. To account for error in the final step of the SIMple-Personalise phase, for example the resident repeatedly mistypes the session id *sid* in Step 9, the link between the UICC and the resident is made tentatively by the issuer until the final phase has succeeded at least once. Our authentication protocols are assumed to run over a secure channel (i.e., using TLS over the standard SA-UA network).

SIMple-OTP provides a standard OTP authentication and runs between the resident R, the UICC U, the requesting entity V and the issuer I. To begin, R has the PIN *pin* and U has the UID *uid* and OTP parameters (k_{OTP}, c). I has private signing key $k_{I\text{sig}}$, the CIDR containing unique residents' UID *uid* and OTP parameters (k_{OTP}, c) linked in the SIMple-Personalise phase.

1. The resident R sends the PIN attempt pin' to the UICC U.
2. If $pin' \neq pin$ then authentication fails. Otherwise U computes the OTP $hotp = \text{HOTP}_{k_{\text{OTP}}}(c)$, increments the OTP counter c, and then sends the authentication message $m_{\text{auth}} = (uid \parallel hotp)$ to V.
3. V sends m_{auth} to the issuer I (e.g., using the SA-UA network).
4. I uses *uid* to look up the resident's eID record of the OTP parameters (k_{OTP}, c) and computes the OTP response $hotp' = \text{HOTP}_{k_{\text{OTP}}}(c)$. If the

OTP is correct, i.e., $hotp' = hotp$, then I computes the response message $m_{\text{resp}} = \textsc{Sign}_{k_{I\text{sig}}}(\text{“}yes''\text{)}$ and increments c. Otherwise, I computes $m_{\text{resp}} = \textsc{Sign}_{k_{I\text{sig}}}(\text{“}no'')$. I sends m_{resp} to the requesting entity V.

5. V verifies the signature on m_{resp} and, if successful and the message is “yes” then authentication succeeds. Otherwise authentication fails.

SIMple-VID provides authentication with improved privacy using public key encryption to hide the resident R's UID *uid* from the receiving entity V. This phase is run between R, the UICC U, V and the issuer I. To begin, R has the PIN *pin* and U has the UID *uid* and OTP parameters (k_{OTP}, c). I has private signing key $k_{I\text{sig}}$, private decryption key $k_{I\text{enc}}$, the CIDR containing unique residents' UID *uid* and OTP parameters (k_{OTP}, c) linked in the SIMple-Personalise phase.

1. The resident R runs Step 1. from the SIMple-OTP phase.
2. If $pin' \neq pin$ then authentication fails. Otherwise U computes the OTP $hotp = \text{HOTP}_{k_{\text{OTP}}}(c)$, encrypts the authentication challenge $c_{\text{chal}} = \textsc{Enc}_{P_{I\text{enc}}}(uid \parallel hotp)$ using the public key of the issuer I and increments the OTP counter c. The UICC U sends c_{chal} to the receiving entity V (i.e., it is shown as a QR code).
3. V runs Step 3. from the SIMple-OTP phase.
4. I recovers *uid* and *hotp* by decrypting c_{chal} using the private encryption key $k_{I\text{enc}}$. Next, I runs Step 4. from the SIMple-OTP phase.
5. V runs Step 5. from the SIMple-OTP phase.

6 Evaluation & Discussion

In this section we evaluate SIMple ID in terms of security and privacy in the foundational eID model. We also present the details of our open-source implementation, benchmark performance and discuss the use of QR codes on basic phone screens.

6.1 Security

The security of de-facto foundational eID authentication is critically dependent on the issuer fulfilling the role of a trusted third party. This is a very strong assumption, and indeed numerous insider attacks have been documented [71], however there are substantive incentives for the issuer to remain honest (but curious [45]). A common motive for less-developed countries to build a national eID platform is minimising fraud, particularly leakages in subsidy programs [58], and therefore maintaining the security of authentication is of primary importance.

It must be emphasised that the SIMple-OTP authentication protocol is simply a standard RFC 4226 HMAC-based OTP [28] accompanying the resident's (independently derived and randomly sampled) UID. A PIN on the user's UICC is used to prevent an adversary with physical device access from generating valid OTPs. The security of HOTP is formally analysed in the standard [57], which shows that no adversary without knowledge of the private key k_{OTP} can do better than approximately brute force. Where d is the number of OTP digits, s is the look-ahead synchronisation window and the adversary is allowed n total attempts, the probability of any adversary succeeding is no greater than $\frac{n*s}{10^d}$ plus some negligible advantage for exploiting the minor algorithmic bias. Whilst the PIN must be kept secret and of sufficient length, the low average transaction values of many foundational eID use cases permits a trade off with usability and convenience.

The SIMple-VID protocol builds on SIMple-OTP, to offer improved privacy from requesting entities, by encrypting the standard HOTP authentication using the public key of the issuer. From a security perspective, even if the encryption algorithm used is wholly insecure, the adversary can still do no better than to try to break the HOTP which is approximately to brute force the private key k_{OTP} as described above.

Compared with the current SMS-based OTP mechanism, security is enhanced by authenticating the UICC hardware rather than the mobile phone number it is linked with. Crucially, this avoids SIM-jacking attacks where a victim's mobile number is switched to a UICC controlled by the adversary [59]. In addition, the use of QR codes means that all 10 available OTP digits can be transmitted without impacting usability; making it around 10,000 times less likely the adversary succeeds in guessing the OTP versus when 6 digits are used.

6.2 Privacy

Firstly, the SIMple-OTP protocol does not offer any privacy benefit compared to the standard foundational eID authentication modalities. We focus therefore on the SIMple-VID protocol which essentially provides a VID-per-transaction functionality. The resident's UID is concealed from requesting entities and the SA-UA network by the use of a secure public key encryption scheme i.e., $c_{\text{chal}} = \textsc{Enc}_{P_{I_{\text{enc}}}}(uid \parallel hotp)$.

The limited cryptographic algorithms natively supported by Java Cards only includes RSA for public key encryption. Though secure when appropriate padding is used [57], RSA encryption suffers from large private key sizes relative to the security provided. The 2048 bit key size recommended for new applications [25] produces 256 byte ciphertexts, necessitating high capacity QR codes with compromised readability on basic phone screens. Fortunately the Elliptic Curve Integrated Encryption Scheme (ECIES) [65] has much smaller ciphertexts for the same level of security and has been reported to operate efficiently on the Java Card platform even without native support [41]. In any case, the lack of a standard implementation led us to prototype SIMple ID using 768 bit RSA encryption. Though insecure for use in a production environment [21], 768 bit RSA is sufficient to demonstrate the required operating principles (i.e., encrypting a digitally signed message) using nonetheless pessimistically high-capacity QR codes.

ECIES is provably secure against adaptive chosen-ciphertext attacks and provides semantic security, as-well as non-malleability [19]. Our application is straightforward and, as noted previously, offers a reduction to the security of HOTP should ECIES be fatally broken in the future.

6.3 Implementation

Our applet implementing the SIMple ID protocols was developed in Java Card and deployed on Taisys SIMoME overlay cards that comply with Java Card 3.0.4 and support the SIM toolkit (Sect. 2.2). To support our proposed extension to the icon protocol and render the QR codes (Sect.‘4), we patched “TTfone TT240“ devices featuring the KaiOS Operating System.

Using these implementations, we now examine whether the latency of our proposed protocols is adequately low for real-life transactions. We focus on SIMple-OTP and SIMple-VID as these are the only two protocols that will be executed repeatedly by the user once the personalization has been completed. We execute each protocol 100 times and measure the runtime of the on-SIM and the on-device parts of the protocol. As seen in Table 1, the total runtime of both protocols is less than 3 seconds in all cases. We observe that SIMple-VID has a longer runtime due to the extra on-SIM operations as well as the increased QR code size. Finally, we note that on-SIM execution is considerably slower than on-device operations. This is expected but highlights the importance of using the SIM only for sensitive cryptographic operations while relying on the phone's CPU for the rest of the computations.

Protocol	**SIM Runtime (ms)**	**Device Runtime (ms)**
SIMple-OTP	2135	220
SIMple-VID	2385	517

Table 1: Average runtime (in milliseconds) of the UICC and the on-device execution components of the SIMple-OTP and SIMple-VID protocols.

6.4 QR codes

Since we display QR codes using basic phones, the screen size and pixel-density are of great importance and place limits on the maximum data capacity and readability. To establish realistic specifications, Amazon was used to identify the ten best selling basic mobile phones in India [13]. We also include the JioPhone, a handset designed to provide 4G internet access for the lowest possible cost. JioPhones are popular in India and Africa where they are heavily subsidised [55]. All of these devices have screens which are either:

- **Basic Screens** characterised by comparatively low pixels-per-inch (ppi) values between 110 and 120. The smallest screen is just 1.5 inches with a 120x120 resolution, although 1.77 inches and 120x160 is the most common.

– **Premium-Basic Screens** all have a resolution of 240x320, a size of 1.77 inches and a pixel-density of 167 ppi.

When considering QR codes that are displayed on such small and low-resolution screens, it is vital to ensure that each module is drawn using as many pixels as possible. Our testing using premium-basic screens found that a minimum module size of 2×2 pixels was required to achieve consistent readability. High levels of ECC proved to be unhelpful as the screens, even when scratched and damaged, provide high acuity compared to the industrial environments QR codes are designed to tolerate. Moreover the reduced net data capacity incurred by sacrificing modules to ECC, and the resulting need for higher version numbers and reduced module sizes, tended to worsen readability. Table 2 shows the maximum module size in pixels-squared (px^2), with an ECC level of 7%, for basic and premium-basic phone screens. Although not shown to save space, premium-basic screens can adequately render a version 13 QR code, providing a capacity of 425 bytes, with a module size of 3 px^2.

Version number	Capacity (bytes)	N. modules	Max. module size 120×120 device (px^2)	Max. module size 240×320 device (px^2)
1	17	21×21	4	8
2	32	25×25	3	7
3	53	29×29	3	6
4	78	33×33	2	5
5	106	37×37	2	5
6	134	41×41	2	4
7	154	45×45	2	4
8	192	49×49	2	4

Table 2: QR code capacity, and maximum module sizes, for the two most common basic mobile phone screen resolutions.

7 Related Work

UICCs for mobile signatures [31], also known as mobile eID [61], are already part of the national eID infrastructure in many countries. Existing solutions are primarily designed for online authentication in developed countries with users that have reliable and affordable cellular connections. Nonetheless, like SIMple ID, these systems also make use of UICCs, mobile devices and existing cellular infrastructure to provide cryptographic identity assurances [31]. Mobile-ID, a typical mobile eID currently deployed in Estonia and Azerbaijan, allows access to e-Government services and digital document signing using a basic mobile phone [60, 62]. In Mobile-ID, the resident provides their mobile number to a website and then a verification code is shown on both on the website and the corresponding mobile phone. If the two codes match the resident enters their secret PIN and the UICC computes a digital signature. Finally, the digital signature is used as an authentication token for the requested website. Mobile-ID has been formally modelled and proven secure using ProVeif [62], but in contrast to SIMple ID may harm privacy by revealing a phone number to the website in every transaction. Similar UICC-based mobile eID are also deployed in many other countries including Finland [76], Moldova [63], Norway [3], Switzerland [68] and Turkey [26]. Zefferer and Teufl [80], and separately Verzeletti et al. [44], systematically review mobile eID systems. Beyond eID, UICCs are widely used for mobile money solutions such as the seminal M-PESA payment service [51]. M-PESA was transformative because it provides a way for those without bank accounts, people marginalised by conventional finance institutions, to send money digitally using only basic mobile phones. Within just two years of its 2007 launch in Kenya, 40% of adults were using the service [27]. Today M-PESA is extensively used for everyday purchases and international remittances by over 51 million customers spanning 7 different African countries [15].

In the academic literature Baqer et al. [23] describe DigiTally, an offline payment system for feature phones based on exchanging short codes between payee and recipient. The usability of DigiTally, another UICC-based CAT application like SIMple ID, is evaluated with participants at a university in Nairobi who report positively upon the usability and perceived security of the system. A notable finding of the DigiTally study, that informed the use of QR codes in this work, is that payers were observed to display their authentication codes to the recipient rather than read them verbally. In related work the authors also evaluate the security and usability tradeoffs of the short authentication codes used for DigiTally [24]. Beyond work focused on less-developed countries, Hassinen and Hyppönen present a protocol which, based on Finland's national PKI register and a Java Card application, provides authentication and non-repudiation using SMS messages [50].

8 Conclusion

SIMple ID can improve the security and privacy properties of existing foundational eID systems without requiring any additional investments in infrastructure. Instead, it employs technologies that the users are already using and are familiar with. Given the scepticism towards biometrics and the pressure for more privacy-preserving systems, we believe that SIMple ID can provide a viable alternative that can reach millions of users. Our techniques furthermore establish a generic platform for displaying QR codes using basic mobile phones that is readily extensible to support new applications such as in finance and targeting aid.

Beyond the technical challenges, a full-scale deployment will require properly incentivised device manufacturers and network operators. In particular, mobile network operators issuing an over-the-air update to their subscribers' SIM cards and handset manufacturers incorporating the icon standard updates. Nonetheless, foundational eID systems are backed by governments who can coordinate the actions required by the different parties. The value proposition is that

Acknowledgments This work was supported, in whole or in part, by the Bill & Melinda Gates Foundation [INV-001309]. Under the grant conditions of the Foundation, a Creative Commons Attribution 4.0 Generic License has already been assigned to the Author Accepted Manuscript version that might arise from this submission. Taisys Technologies Co. Ltd kindly donated 6 SIMoME overlay UICCs and provided technical support.

References

1. Japan Patent JP4258794A. Two-dimensional code having rectangular region provided with specific patterns to specify cell positions and distinction from background, DENSO Wave Corporation (1994)
2. GlobalPlatform Card Specification. Version 2.2.1, GlobalPlatform Inc. (2011)
3. Norwegian Mobile Bank ID: Reaching Scale through Collaboration, GSM (2014)
4. Compendium of Regulations, Circulars & Guidelines for (Authentication User Agency (AUA)/E-KYC User Agency (KUA), Authentication Service Agency (ASA) and biometric device provider) (2018), https://uidai.gov.in/images/resource/compendium_auth_19042018.pdf
5. Understanding Cost Drivers of Identification Systems (2018), https://openknowledge.worldbank.org/bitstream/handle/10986/31065/Understanding-Cost-Drivers-of-Identification-Systems.pdf
6. Aadhaar enrollment / correction / update form. Online, Government of India (2020), https://uidai.gov.in/images/aadhaar_enrolment_correction_form_version_2.1.pdf
7. Commercial National Security Algorithm (CNSA) Suite. MFS U/00/814670-15, National Security Agency (2021)
8. ID systems analysed: Aadhaar. Online, Privacy International (2021), https://privacyinternational.org/case-study/4698/id-systems-analysed-aadhaar
9. Regulation (EU) 2021/953. Official Journal of the European Union L211/1 (2021)
10. Security analysis of the KaiOS feature phone platform for DFS applications. Online, Financial Inclusion Global Initiative, Security Infrastructure and Trust Working Group (2021), https://figi.itu.int/wp-content/uploads/2021/04/Security-analysis-of-the-KaiOS-feature-phone-platform-for-DFS-applications-1.pdf
11. Aadhaar Dashboard. Online, Unique Identification Authority of India (2022), https://uidai.gov.in/aadhaar_dashboard/index.php
12. About MOSIP, Modular Open Source Identity Platform. Online, Modular Open Source Identity Platform (2022), https://mosip.io/mosip/uploads/files/ABOUT%20MOSIP.pdf
13. Amazon.in Bestsellers: The most popular items in Basic Mobiles. Online, Amazon.in (2022), https://www.amazon.in/gp/bestsellers/electronics/1805559031
14. Daily Authentication Transaction Trend, Aadhaar Dashboard (2022), https://uidai.gov.in/aadhaar_dashboard/auth_trend.php?auth_id=dailytrend, Note: 71,477,653,961 Total Authentication Transactions, 53,639,637,282 fingerprint-based.
15. M-Pesa – Africa's leading fintech platform – marks 15 years of transforming lives. Online, Vodaphone Group (2022), https://www.vodafone.com/news/inclusion/mpesa-marks-15-years
16. MOSIP ID Object Definition. Online, Modular Open Source Identity Platform (2022), https://docs.mosip.io/1.1.5/modules/registration-processor/mosip-id-object-definition

17. Population, total - Sub-Saharan Africa. Online, World Bank (2022), `https://data.worldbank.org/indicator/SP.POP.TOTL?locations=ZG`, Note: 1.14 billion indicated population of Sub-Saharan Africa
18. The Mobile Economy 2022. Online, GSM Association (2022), `https://www.gsma.com/mobileeconomy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf`
19. Abdalla, M., Bellare, M., Rogaway, P.: The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In: Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference (2001)
20. Agrawal, S., Banerjee, S., Sharma, S.: Privacy and Security of Aadhaar: A Computer Science Perspective. Economic and Political Weekly (2017)
21. et al, T.K.: Factorization of a 768-Bit RSA Modulus. In: Advances in Cryptology – CRYPTO 2010 (2010)
22. Assisi, C., Ramnath, N.: The Aadhaar Effect: Why the World's Largest Identity Project Matters. Oxford University Press (2018)
23. Baqer, K., Anderson, R., Mutegi, L., Payne, J.A., Sevilla, J.: DigiTally: Piloting Offline Payments for Phones. In: Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017). USENIX Association (2017)
24. Baqer, K., Bezuidenhoudt, J., Anderson, R., Kuhn, M.: SMAPs: Short Message Authentication Protocols. In: Security Protocols XXIV (2017)
25. Barker, E.: Recommendation for Key Management: Part 1 – General. NIST Special Publication 800-57 Part 1 Revision 5 (2020)
26. Birch, D.: Identity Is The New Money. London Publishing Partnership (2014)
27. Camner, G., Pulver, C., Sjöblom, E.: What Makes a Successful Mobile Money Implementation? Learnings from M-PESA in Kenya and Tanzania, GSM (2013)
28. David M'Raihi et al.: HOTP: An HMAC-Based One-Time Password Algorithm. RFC 4226, The Internet Society (2005)
29. Delaporte, A., Bahia, K.: The State of Mobile Internet Connectivity 2021. Tech. rep., GSM Association (2021)
30. Edsbäcker, P.: SIM cards for cellular networks. An introduction to SIM card application development. B.Sc. Thesis, Mid Sweden University (2012)
31. ETSI TR 102 203: Mobile Commerce (M-COMM); Mobile Signatures; Business and Functional Requirements. V1.1.1 (2003)
32. ETSI TS 101 476: Digital cellular telecommunications system (Phase 2+); GSM API for SIM toolkit stage 2 (3GPP TS 03.19 version 8.5.0 Release 1999) (2002)
33. ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 17). V17.1.0 (2022)
34. ETSI TS 102 223: Smart Cards; Card Application Toolkit (CAT). V15.3.0 (2019)
35. ETSI TS 102 226: Smart Cards; Remote APDU structure for UICC based applications (Release 16). V16.0.1, European Telecommunications Standards Institute (2020)
36. ETSI TS 102 384: Smart Cards; UICC-Terminal interface; Card Application Toolkit (CAT) conformance specification (Release 11). V11.0.0 (2022)
37. ETSI TS 131 102: Characteristics of the Universal Subscriber Identity Module (USIM) application (3GPP TS 31.102 version 17.5.0 Release 17) (2022)
38. ETSI TS 131 130: (U)SIM Application Programming Interface (API); (U)SIM API for Java^{TM} Card (3GPP TS 31.130 version 17.0.0 Release 17) (2022)
39. ETSI TS 151 011: Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface (3GPP TS 51.011 version 4.15.0 Release 4) (2005)

40. Ford, B.: Identity and Personhood in Digital Democracy: Evaluating Inclusion, Equality, Security, and Privacy in Pseudonym Parties and Other Proofs of Personhood. arXiv (2020), https://arxiv.org/abs/2011.02412
41. Gayoso Martínez, V., Hernández Encinas, L., Sánchez Ávila, C.: Java Card Implementation of the Elliptic Curve Integrated Encryption Scheme Using Prime and Binary Finite Fields. In: Computational Intelligence in Security for Information Systems (2011)
42. Gelb, A., Metz, A.: Identification Revolution: Can Digital ID be Harnessed for Development? Brookings Institution Press (2018)
43. George, N.A., McKay, F.H.: The Public Distribution System and Food Security in India. International journal of environmental research and public health (2019)
44. Glaidson M. Verzelettiet et al.: A National Mobile Identity Management Strategy for Electronic Government Services. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (2018)
45. Goldreich, O.: Foundations of Cryptography, vol. 2. Cambridge Uni. Press (2004)
46. GS1 General Specifications: The foundational GS1 standard that defines how identification keys, data attributes and barcodes must be used in business applications. Release 22.0, GS1 (2022)
47. GSM 11.11: Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module – Mobile Equipment interface;. V5.3.0 (1996)
48. Gupta, B., Quamara, M.: A taxonomy of various attacks on smart card–based applications and countermeasures. In: Concurrency and Computation: Practice and Experience (2021)
49. Handschuh, H., Paillier, P.: Smart Card Crypto-Coprocessors for Public-Key Cryptography. In: Smart Card Research and Applications (2000)
50. Hassinen, M., Hypponen, K.: Strong Mobile Authentication. In: 2005 2nd International Symposium on Wireless Communication Systems (2005)
51. Hughes, N., Lonie, S.: M-PESA: Mobile Money for the "Unbanked" Turning Cellphones into 24-Hour Tellers in Kenya. Innovations: Technology, Governance, Globalization (2007)
52. ISO/IEC 18004:2015: Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification (2015)
53. ISO/IEC 7816-4:2020: Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange (2020)
54. Ivatury, G., Mas, I.: The Early Experience with Branchless Banking. Focus Note No. 46, CGAP (2008)
55. James, J.: The smart feature phone revolution in developing countries: Bringing the internet to the bottom of the pyramid. The Information Society (2020)
56. Java Card Platform: Runtime Environment Specification. Version 2.2.1 (2003)
57. Kaliski, B., Staddon, J.: PKCS #1: RSA Cryptography Specifications Version 2.0. RFC 2437, The Internet Society (1998)
58. Khera, R.: Impact of Aadhaar in Welfare Programmes. Economic and Political Weekly (2017)
59. Konoth, R.K., Fischer, B., Fokkink, W., Athanasopoulos, E., Razavi, K., Bos, H.: SecurePay: Strengthening Two-Factor Authentication for Arbitrary Transactions. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P) (2020)
60. Krimpe, J.: Mobile ID: Crucial Element of m-Government. In: Proceedings of the 2014 Conference on Electronic Governance and Open Society: Challenges in Eurasia. Association for Computing Machinery (2014)
61. Kubach, M., Leitold, H., Roßnagel, H., Schunck, C.H., Talamo, M.: SSEDIC.2020 on Mobile eID. In: Open Identity Summit 2015 (2015)

62. Laud, P., Roos, M.: Formal Analysis of the Estonian Mobile-ID Protocol. In: Identity and Privacy in the Internet Age (2009)
63. Manoil, V., Turcanu, I.: Moldova Mobile ID Case Study, World Bank (2018)
64. Martin, A.K.: Aadhaar in a Box? Legitimizing Digital Identity in Times of Crisis. Surveillance & Society (2021)
65. Martínez, V.G., Álvarez, F.H., Encinas, L.H., Ávila, C.S.: A comparison of the standardized versions of ECIES (2010)
66. Mavroudis, V., Svenda, P.: Jcmathlib: Wrapper cryptographic library for transparent and certifiable javacard applets. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P) Workshops (2020)
67. MOSIP Docs 1.2.0: ID Authentication Services, Modular Open Source Identity Platform (2022), `https://docs.mosip.io/1.2.0/modules/id-authentication-services`
68. Murphy, A.: Swisscom Mobile ID: Enabling an Ecosystem for Secure Mobile Authentication, GSM Association (2018)
69. Naumann, I., Hogben, G.: Privacy features of European eID card specifications. Network Security (2008)
70. Parsovs, A.: Estonian Electronic Identity Card: Security Flaws in Key Management. In: Proceedings of the 29th USENIX Conference on Security Symposium (2020)
71. Qin, K., Zhou, L., Livshits, B., Gervais, A.: India's "Aadhaar" Biometric ID: Structure, Security, and Vulnerabilities. In: Financial Cryptography and Data Security – 26th International Conference (2022)
72. Rajput, A., Gopinath, K.: Analysis of Newer Aadhaar Privacy Models. In: Information Systems Security (2018)
73. Reaves, B., Scaife, N., Bates, A., Traynor, P., Butler, K.R.B.: Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. In: 24th USENIX Security Symposium (2015)
74. Reid, J., Looi, M.: Making Sense of Smart Card Security Certifications. In: Smart Card Research and Advanced Applications: IFIP TC8 / WG8.8 Fourth Working Conference on Smart Card Research and Advanced Applications (2000)
75. Salem, A.M., Elhingary, E.A., Zerek, A.R.: Value added service for mobile communications. In: 4th International Conference on Power Engineering, Energy and Electrical Drives (2013)
76. Trichina, E., Hyppönen, K., Hassinen, M.: SIM-enabled Open Mobile Payment System Based on Nation-wide PKI. In: ISSE/SECURE 2007 (2007)
77. Vashistha, A., Anderson, R., Mare, S.: Examining Security and Privacy Research in Developing Regions. In: Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies. COMPASS '18 (2018)
78. Švenda, P.: Nuances of the JavaCard API on the cryptographic smart cards – JCAlgTest project. In: 7th International Workshop on Analysis of Security API (2014)
79. Wong, C.W.T., Tsui, T.C.: Automated Payment over the Counter – A study of Alipay, WeChat Wallet and Octopus currently used in Mainland China and Hong Kong. In: The Future of the Commercial Contract in Scholarship and Law Reform Fourth Annual Conference, Institute of Advanced Legal Studies (2019)
80. Zefferer, T., Teufl, P.: Leveraging the Adoption of Mobile eID and e-Signature Solutions in Europe. In: Electronic Government and the Information Systems Perspective (2015)

All links were last followed on May 30, 2022 and have been archived at `https://web.archive.org`.